

Wahrheiten über Ihre IT, die Sie gar nicht hören wollen.

Stefan Berner
Diso AG
Gümligen / Bern

Schlüsselworte

Projektmanagement, Softwarequalität

Einleitung

Softwareentwicklung führt zu oft in ein Fiasko. Unsere Branche glänzt nur selten mit reifen Ingenieurleistungen und zufriedenen Kunden. Hier sind einige *Wahrheiten* zusammengefasst, die wir alle irgendwie kennen, die wir aber lieber nicht öffentlich diskutieren wollen. Die Konsequenzen die sich aus einer ernsthaften Beschäftigung mit diesen Themen ergeben, möchten wir lieber nicht (jedenfalls nicht gerade jetzt) auf uns nehmen.

Diese Kompilation von Wahrheiten steht unter dem Motto:

Jede Ähnlichkeit mit lebenden Personen (oder realen Firmen) haben sich diese selber zuzuschreiben.

Der Schweizer Liedermacher Mani Matter pflegte diesen Spruch (ohne die Klammer) vor seinem Lied über Politiker zu sagen.

Teil 1: TCO

Gespart muss ein, koste es was es wolle.

Das Problem

Warum wird wider besseren Wissens so viel qualitativ schlechte Software erstellt? Warum werden so viele Projekte in den Sand gesetzt? Was hält Manager, Projektleiter und Entwickler davon ab ihre Arbeit richtig zu machen? Und, was heisst überhaupt richtig?

Ich bin überzeugt, die Gesamtkosten (Total Cost of Ownership, TCO) sind eines der besten, wenn nicht sogar das einzig zulässige Qualitätskriterium für Software.

Jeder Fehler, jedes Ärgernis, jede Dateninkonsistenz und Projektverzögerung schlägt sich letztendlich direkt oder indirekt in den Gesamtkosten nieder. Sei es in Form von direkter Arbeit, sei es als Rufschädigung oder als hohe Fluktuation von Kunden und Mitarbeitenden. Die jahrzehntelange Erfahrung in der IT hat mich gelehrt dass Software vor allem dann stabil und gut wird, wenn jede Projektaufgabe in jeder Phase richtig und vollständig gelöst wird. Richtig im Sinne von: alle Massnahmen,

die die Qualität verbessern helfen, werden konsequent umgesetzt. Jede Auslassung, jede Schwachstelle, jede Unschönheit im Design, jedes "das können wir dann später noch richtigstellen" rächt sich früher oder später. Meistens früher.

Die Ursache

Was hindert uns daran immer die beste Qualität zu liefern? Wird in der Entwicklungsphase die Zeit knapp, so hört man oft Sätze wie:

Wir wissen schon, dass die Anforderung noch nicht vollständig ist aber,...

Eine Datenübernahme wäre ein guter Zeitpunkt um die Datenqualität zu verbessern, aber ...

Die Dokumentation ist nicht ganz korrekt, aber ...

Ich bin sicher Sie finden selber weitere Beispiele aus ihrem Bereich. Dabei kann der Aber-Nebensatz wahlweise durch

...Time-to-market

...wir müssen endlich vorwärts kommen

...100%tige Sicherheit wird man auch so nicht erreichen

...wer soll dies bezahlen?

...wie kann ich das gegenüber dem Management begründen?

ergänzen.

All diese Ausreden sind Ausdruck von Hilflosigkeit und Überforderung. Sie sind der Versuch, die eigene Unzulänglichkeit und frühere Fehler zu kaschieren. Es ist absolut verständlich dass niemand gerne zugibt, dass die alte Lösung auf einer falschen Architektur basiert, dass die Qualifikation der Programmierer ungenügend war oder dass man die Benutzer zu wenig einbezogen hatte. Ebenso nachvollziehbar ist, dass sich niemand gerne vor seine Auftraggeber hinstellt und beichtet, das Problem sei komplexer als angenommen, denn man hätte die Anforderung nicht wirklich verstanden, oder man hätte den Aufwand hoffnungslos unterschätzt. Bei allem Verständnis für die persönliche Situation der Betroffenen, der Sache, sprich die möglichst kleinen TCO, ist damit nicht geholfen.

Die Lösung

Fokussieren Sie ihre Aufmerksamkeit auf die TCO. Fragen Sie bei Projektbeginn nach den TCO. Fragen Sie bei jeder Massnahme nach ihrem Einfluss auf die TCO.

Lassen Sie sich nicht durch selbst auferlegte und nicht-zwingende Restriktionen davon abbringen, jede Aufgabe richtig zu erledigen. Der allgemeingültige Spruch "jede Arbeit die es wert ist gemacht zu werden, ist es wert richtig gemacht zu werden" gilt auch für die Informatik. Gehen Sie davon aus, dass jede Schwachstelle irgendwann bricht. Faktoren wie Time-to-market, Budgetperioden, Bonussicherung, (firmen-)politische Befindlichkeiten und die Verfügbarkeit von Ressourcen sind Argumente, die in eine Projektbearbeitung einfließen müssen. Geben Sie diesen Argumenten die Gewichtung welche ihnen im Rahmen der TCO zusteht. Es sind Faktoren für die Entscheidungsfindung und die Vorgehensplanung. Es sind aber keine unverrückbaren "Tatsachen" um Entscheide zu rechtfertigen.

Auch Auftraggeber und Kontrollorgane können und müssen mithelfen die Qualität in allen Phasen zu verbessern. Lassen Sie sich nicht von "technischen" Begründungen

dazu verführen, Probleme auszusitzen. Misstrauen Sie den Standardausreden wie "Wir vertrauten den (externen) Spezialisten.", "Das Controlling hat versagt.", "Andere Firmen haben auch diese Lösung gewählt." Denken Sie selber mit. Rechnen Sie über den Einführungstermin hinaus. Was kostet mich als Hersteller ein neues Automodell, fertig gestellt pünktlich zum jährlichen Autosalon, wenn ich es nach einem Jahr zurückrufen muss?

Sie wollen qualitativ gute Software?

Hören Sie auf Project Management by Budget cut zu unterstützen.

Hören Sie auf Project Management by Controlling zu zelebrieren.

Lösen Sie jede einzelne Aufgabe richtig, und vollständig.

Teil 2: Benutzereinbezug

Wer nicht sagt was er braucht, bekommt was er verdient.

Das Problem

Benutzereinbezug ist wichtig". Ich habe noch nie etwas Gegenteiliges gehört. Doch trotz diesem Unisono-Bekenntnis gibt es ihn kaum.

Haben Sie Folgendes schon gehört?

- Benutzer sagen nie richtig, was sie wollen".
- Die können die komplexen Modelle sowieso nicht verstehen".

Oder von der Gegenseite:

- Die haben uns gar nie gefragt".
- Das ist uns zu technisch, das müssen wir nicht verstehen".
- Das sind IT-Themen, dafür seid ihr zuständig".

Dies sind alles hilflose Sprüche, welche die Kommunikationsprobleme zwischen IT Anwendern und Herstellern/Entwicklern verdeutlichen.

Die Ursache

Fachleute einerseits, welche sich Informatik Grundkenntnisse angeeignet haben, sowie Informatiker andererseits, welche kaum je ein IT Projekt im relevanten Fachgebiet bearbeitet haben, wissen meistens zu wenig über die jeweils andere Disziplin.

Kenntnisse der Theorie oder Allgemeinwissen über ein Fachgebiet reichen nicht für eine gute Kommunikation. Gute Software ist auf die Verwendung in einem konkreten Fachbereich zugeschnitten. Detailhandel ist nicht gleich Detailhandel, Lagerverwaltung nicht gleich Lagerverwaltung, Patientendossier nicht gleich Patientendossier.

IT Spezialisten haben nicht den Ruf, in nicht-technischer Kommunikation zu glänzen. Ihre Bereitschaft mit den Fachanwendern über deren Sicht der Welt zu diskutieren hält sich in Grenzen. "Ich brauche mir doch nicht erklären zu lassen, was ein Artikel ist. Schliesslich habe ich schon selber welche gekauft. Ein Artikel ist eh immer dasselbe."

Sobald sie glauben etwas verstanden zu haben, erstellen sie Programmvorgaben und technische Dokumente - und von da an ist die Sache geritzt. Detailprobleme werden ad hoc gelöst, für grössere hat man keine Zeit oder es wird die "da müssten wir ganz von vorne anfangen, das wollen Sie sicher nicht, oder?" - Karte ausgespielt. Im Fachjargon nennt man das ein Problem wegdiskutieren. Leider ist das Problem nicht weg, und spätestens im Betrieb merkt der Kunde dass er sich über den Tisch hat ziehen lassen.

Auf der anderen Seite befinden sich die Kunden, die Auftraggebenden (Fach- und Management-Ebene) die in Bezug auf Kommunikation über technische Belange

mindestens so wenig Affinität mitbringen wie die IT-Leute für die Fachseite. IT-Projekte sind meist komplex und ihr Verständnis verlangt ein hohes Mass an Abstraktionsvermögen. Sich damit ernsthaft zu beschäftigen erfordert Denkarbeit.

Wer auf das Prinzip *Hoffnung* setzt, à la...

- Die werden wohl wissen was zu tun ist"
- Beim 3. Anlauf muss es ja klappen"
- Die Standardsoftware hat ein *xy module*, damit sind doch all unsere Anforderungen abgedeckt.")

der verdient es nicht besser!

Die Lösung

Es gibt kein erfolgreiches IT-Projekt, wenn sich Hersteller/Entwickler und Anwender nicht ausgetauscht und verstanden haben. Benutzereinbezug bedeutet echte Kommunikation zwischen den Beteiligten. Er bedingt gegenseitiges Verstehen. Ohne dieses brauchen Sie ihr Projekt gar nicht zu beginnen. Sie werden scheitern.

Kunden unterscheiden sich in Arbeitsweise, Grösse, Kundensegment, Standort, Produktesegment, und so weiter. Erst wenn auch diese spezifischen, für jedes Unternehmen einmaligen Aspekte in die Software einbezogen werden, wird sie brauchbar und erfolgreich. Bei Standardsoftware heisst das dann "Customizing", was ohne funktionierende Kommunikation und echtes Verständnis ebenfalls nicht zielführend ist.

Benutzereinbezug heisst, dass das kombinierte Wissen der Fachleute und der IT in die Arbeit einfließt. Nur so wird ein Projekt erfolgreich.

Für Informatiker in Schlüsselrollen (Architekt, Projektleiter) führt kein Weg daran vorbei, sich in die Welt des Kunden einzudenken. Die spezifischen Umstände, die Kultur, die eingespielten Arbeitsabläufe wirklich zu verstehen. Kombinierte Ausbildungen wie Wirtschaftsinformatik oder Medizinalinformatik sind gute Voraussetzungen um sich in der Welt der Kunden zurechtzufinden. Sie sind aber noch keine Garantie, dass man genau jenen Kunden auch versteht.

Als Auftraggeber müssen Sie sich einbeziehen lassen. Sie müssen sich die Mühe machen, Modelle und Ergebnisse der Informatiker zu verstehen. Was natürlich voraussetzt, dass diese verständliche Dokumente liefern.

Benutzereinbezug bedeutet harte Arbeit für beide Seiten. Natürlich ist das Prinzip Hoffnung einfacher. Die Statistik der nicht erfolgreichen Projekte spricht leider gegen letzteres.

Sie wollen qualitativ gute Software?

- Investieren Sie in Kommunikationskultur.
- Übergeben Sie die Spezifikationsarbeiten an der Schnittstelle zwischen Fach- und IT Leuten mit guten Kommunikationsfähigkeiten.
- Akzeptieren Sie als Auftraggeber kein Dokument...
...das Sie nicht wirklich verstanden haben...
...mit dessen Inhalt Sie nicht 100% einverstanden sind...
...dessen Konsequenzen auf das Gesamtprojekt Sie nicht abschätzen oder nachvollziehen können....

- Akzeptieren Sie keine Ausreden, warum die Konzepte und Modelle nicht von allen verstanden werden (können).
- Fragen Sie nicht "Habt Ihr die Fachabteilung gefragt?" oder "Habt Ihr das IT-Konzept gelesen?". Fragen Sie "Habt Ihr euch gegenseitig verstanden?"

Teil 3: Management Attention

Wie die Leitung, so die Leistung.

Das Problem

Management attention ist ein kritischer Erfolgsfaktor. Das steht in jedem Handbuch über IT-Projekte. Es ist eine der Weisheiten, die jeder bekräftigt und kaum einer versteht geschweige denn umsetzt. In vielen Projekten wird die Aufmerksamkeit der Vorgesetzten erst erreicht, wenn Zeit- oder Geld-Budgets überschritten sind. Nicht selten übernimmt dann ein Mitglied der Geschäftsleitung das Ruder. Es werden "griffige Massnahmen" eingeleitet und das Projekt geht immer noch bzw. noch schneller schief. Das nennt man dann management attention.

Die Ursache

Was genau ist gemeint mit management attention? Es ist nicht damit getan, dass die Geschäftsleitung regelmässig einen geschönten Projektbericht erhält, dessen Kurzform von folgender generischer Natur ist: "Es gibt einige Probleme, Massnahmen wurden ergriffen, wir sind immer noch auf Kurs und zuversichtlich, dass wir die Ziele erreichen können."

Es ist auch nicht damit getan, dass das Management in Krisensituationen mehr Geld und Leute organisiert.

Der Einfluss von Softwareprojekten auf die Firmenkultur und den Geschäftsgang wird häufig unterschätzt. Softwaresysteme geben vor, wie in den Fachabteilungen gearbeitet wird. Geschäftsprozesse, Regeln etc. werden durch die Software nicht nur unterstützt sondern definiert, erzwungen oder eingeschränkt. Jeder falsche oder unflexible Geschäftsprozess, jede Inkonsistenz in zentralen Geschäftsdaten kann zu hohen Verlusten führen. Das kann jeder bestätigen der (Standard-)Software eingeführt hat und auf die Auswirkungen auf Firmenkultur und Arbeitsabläufe schlecht vorbereitet war.

Die Lösung

Jedes grössere Projekt braucht ein Mitglied des höheren Managements als Product Owner. Dies ist kein Ehrenamt, sondern eine ernstzunehmende, verantwortungsvolle Aufgabe. Es braucht jemanden der oder die nicht nur Gelder spricht, sondern sich so weit in das Projekt eindenkt und einbringt, dass er oder sie weiss, worauf sich die Firma einlässt.

Es genügt nicht, sich mit den Lieferanten auf Ziele und Absichten zu verständigen. Es genügt nicht, sich von seinen Zukunftskollegen bestätigen zu lassen, sie hätten die selbe Software und die selben Probleme.

Es braucht jemanden im Projekt, der für die Firma mitdenkt. Das ist nicht gratis zu haben. Es braucht neben Zeit auch persönliches Engagement. Die Standardausrede "Wir können es uns nicht leisten, uns um solche Details zu kümmern." heisst in der

Realität "Wir können es uns nicht leisten, uns *nicht* um solche Details zu kümmern."

Sie wollen qualitativ gute Software?

- Bestimmen Sie ein Mitglied des oberen Managements als aktiven Paten für das Projekt.
- Wenden Sie als verantwortlicher Manager genügend Zeit und Energie auf, um sich in das Projekt einzuarbeiten. Sie müssen wissen was das Projekt bzw. sein Scheitern für die Firma bedeuten.
- Stellen Sie sicher, dass die Einbettung in und die Auswirkungen auf Ihre Firma bei Auftraggeber und -nehmer bekannt sind.
- Stellen Sie sicher, dass Firmenkultur und wichtige Geschäftsprozesse gebührend in das Softwareprojekt einfließen.
- Formulieren Sie Ihre Projektziele als überprüfbare und sanktionierbare Anforderungen. Machen Sie diese zum Gegenstand des Auftrages bzw. Vertrages.

Teil 4: Qualifikation der Informatiker

... denn sie wissen nicht was sie tun.

Das Problem

Es ist gut erforscht, warum IT Projekte Erfolg haben oder eben nicht. Zu den wichtigsten Faktoren gehört die Qualifikation der Personen in den Schlüsselpositionen. Schlüsselpositionen sind Projektleitung, System- und Softwarearchitektur.

Was heisst Qualifikation? Glaubt man den Anforderungen in grossen (WTO-) Ausschreibungen, ist jemand qualifiziert, wenn er oder sie Erfahrung sowie eine Ausbildung für das Fachgebiet oder die spezifische IT-Umgebung hat. Der Glaube, dass eine Ausbildung ein Garant für Fähigkeiten ist, führt dazu, dass viele Leute falsch eingesetzt werden und überfordert sind.

Die Ursache

Fähigkeiten sind einerseits Ausbildung, Fachwissen und Erfahrung. Diese Aspekte sind relativ einfach prüfbar. Andererseits ist eine adäquate Denkfähigkeit (Intelligenz) eine unabdingbare Voraussetzung um eine komplexe Aufgabe überblicken und lösen zu können. Intellektuelle Fähigkeiten sind in unserer Gesellschaft ein Tabuthema. Entsprechend schwer tun sich viele Verantwortliche damit.

Meiner Erfahrung nach ist ein Grossteil der Beteiligten in Softwareprojekten von ihren Aufgaben überfordert. Häufig liegt es nicht an fehlender Erfahrung oder fehlendem Fachwissen für die eigentliche Aufgabe. Es liegt daran, dass die Leute nicht entsprechend ihrer Fähigkeiten eingesetzt werden.

Die Überforderung beginnt bei Business Analysten und Architekten, welche die fachlichen Probleme nicht wirklich verstehen oder überblicken. Sie liefern schlecht dokumentierte, unvollständige, falsche Lösungen und Modelle. Es geht weiter mit Projektleitern, welche durch die Komplexität der Organisationsaufgaben intellektuell überfordert sind.

Aus dieser Überforderung resultieren Planungssünden. Diese müssen in den weiteren Phasen korrigiert werden. Diese Zusatzbelastung macht die Folgeaufgaben (Programmierung, Datenübernahme, Installation) viel komplexer, was die Programmierer und Systemtechniker überfordert. Dies schlägt sich in der Qualität der Ergebnisse nieder.

Nach meiner Hypothese korreliert die Qualität eines Softwareprojektes mit dem IQ der Personen in den Schlüsselpositionen.

Die Lösung

Um ein derart komplexes System wie ein grosses IT Projekt überblicken, begreifen, planen und führen zu können braucht es sicher Fachwissen und Erfahrung. Weiter braucht es auch Fähigkeiten wie Abstraktionsvermögen, Sozialkompetenz,

analytisches Denken, kommunikative Fertigkeiten, schnelle Auffassungsgabe und psychologisches Geschick. Fähigkeiten, die alle mit dem Intelligenzquotienten positiv korrelieren. Kurz es braucht einen hohen IQ.

Bei grösseren Projekten braucht es in den Schlüsselpositionen Projektleitung und Architektur Leute mit einem überdurchschnittlichen IQ. Meiner persönlichen Einschätzung nach ein IQ in den höchsten 10% (IQ 120+). So bestehen gute Chancen, dass die Leute nicht überfordert sind und die Gesamtaufgabe überblicken und organisieren können. Als Folge wird die Umsetzung einfacher. Dies führt dazu, dass die Leute, welche das System bauen ihren Aufgaben gewachsen sind. Damit erhöhen sich die Chancen massiv, das Projekt zu einem Erfolg zu machen.

Sie wollen qualitativ gute Software?

- Stellen Sie sicher, dass die entscheidenden Posten im Projekt mit ausreichend intelligenten Leuten besetzt sind.
- Prüfen Sie nebst Erfahrung und Fachwissen den IQ als Qualifikation.
- Stellen Sie sicher, dass jede Person entsprechend ihren Fähigkeiten (Erfahrung, Fachwissen und Intelligenz) eingesetzt wird und dass die Aufgaben tatsächlich ihrem Profil entsprechen.

Teil 5: Persönliches Engagement

Wenn Du ein Schiff bauen willst, dann trommle nicht Männer zusammen um Holz zu beschaffen, Aufgaben zu vergeben und die Arbeit einzuteilen, sondern lehre die Männer die Sehnsucht nach dem weiten, endlosen Meer".

Antoine de Saint-Exupéry

Das Problem

Projekte haben mit Menschen zu tun. Nutzende, Auftraggebende, Entwickelnde, Projektleitende. Es braucht die gemeinsame Anstrengung aller Beteiligten um ein Projekt erfolgreich zu machen. Qualität wird erschaffen, sie lässt sich nicht administrieren!

Damit ein Team gut zusammenarbeitet, braucht es mehrere Faktoren: Arbeitsbedingungen, "Chemie" zwischen den Team-Mitgliedern, Teamgeist etc.

Ein wichtiger Faktor ist meiner Ansicht nach das persönliche Engagement möglichst vieler beteiligter Personen. Ein Softwareprojekt kann nur gut gehen, wenn sich jemand mit Herzblut dafür engagiert.

Die Ursache

In einer typischen Projektplanung läuft zu viel über messbare Ressourcen und technische Details. "Wer ist der billigste Anbieter", "Wer kann am schnellsten liefern?", "Welche Programmierumgebung ist die beste?", "Welches Datenbanksystem ist am günstigsten?"

Es ist schwieriger zu prüfen, ob die Leute geeignet sind (siehe Blog 4, Qualifikation der Informatiker) und mit welcher Einstellung sie die Aufgaben wahrnehmen. Leute die ihren Job nur ausführen weil sie dafür bezahlt sind, Leute die möglicherweise unter Margendruck nur die Minimalleistung erbringen (dürfen) sind bestenfalls als Hilfskräfte für klar definierte Aufgaben einsetzbar. Sie sind nicht die Leute die den Karren aus dem Dreck ziehen, die das Projekt zum Erfolg führen.

Es gibt zu viele Leute die sich dem Auftrag statt der Aufgabe verpflichtet fühlen.

Persönliches Engagement ist nicht mit Ehrgeiz zu verwechseln. Wer ein erfolgreiches Grossprojekt braucht um die eigene Karriere zu fördern, ist die falsche Person. Die Motivation muss sein: "ich will etwas Gutes, Brauchbares, Nützliches schaffen".

Die Lösung

Jedes Projekt braucht mindestens einen Macher, eine Macherin.

Jemanden, der oder die

- für das Projekt (ein-)steht.
- mit persönlichem Engagement das Projekt zum Erfolg führt.
- von allen Beteiligten im Projekt XY als Missis-XY oder Mister-XY erkannt wird.

- das Projekt persönlich nimmt, den eigenen Ruf daran hängt und ehrlich glaubt, dass der eingeschlagene Weg zum Erfolg führt.

Es ist nicht entscheidend, in welcher Rolle diese Person im Projekt mitmacht. Mit Vorteil hat sie Einsitz in einem Steuerungsgremium (Projektausschuss, Projektleitung, Architektur-Board, etc.). Es kann jemand aus dem Management, aus dem Fachbereich oder aus der IT-Truppe sein. Am besten ist ein Projekt bedient, wenn aus jedem Bereich jemand persönlich engagiert ist.

Diese Person wird persönlich darüber wachen, dass erfolgsgefährdende Massnahmen abgewendet werden. Sie wird dafür sorgen, dass notwendige Qualitätsvorgaben befolgt werden. Sie wird sich gegen politische und budgetäre Restriktionen zur Wehr setzen. Nicht weil es ihre Pflicht ist, weil sie will, dass das Projekt ein Erfolg wird.

Sie wollen qualitativ gute Software?

- Starten Sie kein Projekt ohne Mister-XY oder Missis-XY. Jemand der oder die das Projekt als persönliche Herausforderung übernimmt.
- Bevorzugen Sie engagierte Leute vor spezialisierten oder billigen Leuten.
- Sorgen Sie dafür, dass auf allen Ebenen (Management, Fachbereich, IT-Erstellung, IT-Betrieb) jemand ist, der oder die dieses Projekt nicht bloss als "weitere Aufgabe" betrachtet.

Kontaktadresse:

Stefan Berner
Diso AG
Morgenstrasse 1
CH-3073 Gümligen

Telefon: +41 79 474 71 41

E-Mail: sberner@diso.ch

Internet: www.diso.ch